WEST Search History

DATE: Sunday, December 14, 2003

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT,PGPB,JPAB; PLUR=YES; OP=OR		
L16	L15 and (floating same point)	13	L16
L15	L13 and ((virtual or mask$) same (interrupt or exception or trap$))	21	L15
L14	L13 and synchronous	15	L14
L13	L12 and block$	31	L13
L12	L11 and l2	32	L12
L11	emulator same handl$ same (exception or trap)	62	L11
L10	L6 and emulator	49	L10
L9	L6 and mask$	27	L9
L8	l3 and L7	2	L8
L7	(operating same system) same emulat$ same embed$	85	L7
L6	L5 and l3	51	L6
L5	(operating same system) same (emulator or embed$)	6022	L5
L4	L3 and (platform or legacy)	54	L4
L3	L2 and l1	150	L3
L2	(operating same system) same emulat$ same (exception or interrupt)	391	L2
L1	exception same handl$	9002	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 13 of 13 returned.

1. Document ID: US 20030135791 A1

AB: A system S is defined which is capable of simulating a computer (virtual computer, VC) for the purpose of software performance monitoring. The system is implemented as a set of software modules (SM) that can be exchanged to change the behavior of the VC. The VC is driven by a CPU emulator, and can run any operating system (virtual operating system, VOS) that is supported by the available SM's. The system is designed to log accesses to system resources and the nature of these accesses. The system is particularly useful for determining whether an executable or file contains an unknown virus, with a very low risk of false positives. Detected viruses include encrypted, polymorphic, metamorphic and other virus types.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

2. Document ID: US 6631514 B1

AB: The inventive emulator dynamically translates instructions in code written for a first architecture into code for a second architecture. The emulator designates various checkpoints in the original code, and speculatively reorders the placement of the translated code instructions according to optimization procedures. If during the execution of the reordered code, a trap should occur, then the emulator resets the original code to the most recent checkpoint and begins executing the original code sequentially in a line-by-line manner until the section is completed or branched out of. The original code is reset by changing the program counter to the checkpoint, and reversing the effects of each instruction which has been executed subsequent to the checkpoint. Thus, any native instructions which correspond to original instructions which occur sequentially prior to the checkpoint have been executed, and any native instructions which correspond to original instructions which occur sequentially subsequent to the checkpoint have not been executed.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

3. Document ID: US 6549959 B1

AB: A method and computer for executing the method. A CPU is programmed to execute first and second processes, the first process programmed to generate a second representation in a computer memory of information of the second process stored in the memory in a first representation. A main memory divided into pages for management by a virtual memory manager that uses a table stored in the memory. DMA (direct memory access) monitoring circuitry and/or software is designed to monitor DMA memory write transactions to a main memory of a computer by a DMA device of the computer; to detect when the first representation is overwritten by a DMA memory write transaction initiated by the second process, without the second process informing the first process of the DMA memory write transaction, the detecting guaranteed to occur no later than the next access of the second representation following the DMA memory write transaction; to record an indication of a location in the main memory written by the DMA memory write transaction, the DMA monitoring circuitry designed to operate without being informed of the DMA memory write transaction by a CPU of the computer before initiation of the DMA memory write transaction, and to provide the indication to the CPU on request; and to report to the first process that the first representation is overwritten by a DMA memory write transaction. The DMA monitoring circuitry includes a plurality of registers outside the address space of the main memory, each register including an address tag and a vector of memory cells, and control circuitry designed to establish an association between a one of the plurality of registers with a region of the memory when a modification to the region is detected by setting the address tag of the one register to an approximation of the address of the region, and to set the values of the memory cells of the vector to record a fine indication of the address of a memory location modified, the control circuitry being operable without continuing supervisory control of a CPU of the computer. Circuitry is designed to record indications of modification to pages of the main memory into the registers. Read circuitry is designed to respond to a read request from the CPU by providing an address of a modified memory location. The virtual memory management tables do not provide backing store for the modification indications stored in the registers.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw Desc | Image

---

☐ 4. Document ID: US 6496847 B1

AB: A virtual machine monitor (VMM) is included in a computer system that has a protected host operating system (HOS). A virtual machine running at least one application via a virtual operating system is connected to the VMM. Both the HOS and the VMM have separate operating contexts and disjoint address spaces, but are both co-resident at system level. A driver that is downloadable into the HOS at system level forms a total context switch between the VMM and HOS contexts. A user-level emulator accepts commands from the VMM via the system-level driver and processes these commands as remote procedure calls. The emulator is able to issue host operating system calls and thereby access the physical system devices via the host operating system. The host operating system itself thus handles execution of certain VMM instructions, such as accessing physical devices.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw Desc | Image

☐ 5. Document ID: US 6397379 B1

AB: A method and a computer for execution of the method. As part of executing a stream of instructions, a series of memory loads is issued from a computer CPU to a bus, some directed to well-behaved memory and some directed to non-well-behaved devices in I/O space. Computer addresses are stored of instructions of the stream that issued memory loads to the non-well-behaved memory, the storage form of the recording allowing determination of whether the memory load was to well-behaved memory or not-well-behaved memory without resolution of any memory address stored in the recording.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw. Desc | Image

☐ 6. Document ID: US 6199152 B1

AB: A method of responding to an attempt to write a memory address including a target instruction which has been translated to a host instruction for execution by a host processor including the steps of marking a memory address including a target instruction which has been translated to a host instruction, detecting a memory address which has been marked when an attempt is made to write to the memory address, and responding to the detection of a memory address which has been marked by protecting a target instruction at the memory address until it has been assured that translations associated with the memory address will not be utilized before being updated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw. Desc | Image

☐ 7. Document ID: US 6173248 B1

AB: A method and apparatus for handling masked exceptions that receives an exception from the operating system on which an emulator is running a user program and determines the origin of the exception. If the emulator generated the exception, the emulator handles the exception internally and returns it to the operating system. If the emulated user program generated the exception, the emulator checks the status of the exception type. If that type of exception is blocked, the exception is marked as deferred. Otherwise, the exception is delivered to the user application or marked as pending for later delivery. The system and method can maintain a virtual exception mask to indicate the status of the exception type.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc
Image

8. Document ID: US 6031992 A

AB: A microprocessor for a host computer designed to execute target application programs for a target computer having a target instruction set including the combination of code morphing software, and morph host processing hardware designed to execute instructions of a host instruction set, the combination of the code morphing software and the morph host processing hardware comprising means to translate a set of target instructions into instructions of a host instruction set, means to optimize the instructions of the host instruction set translated from the target application program speculating upon the occurrence of a condition, means to determine under control of the code morphing software official state of the target computer which existed at the beginning of a translation of a set of target instructions during execution of the target application program by the microprocessor, means for updating state of the target computer from state of the host computer when a set of host instructions executes in accordance with the speculation, means to detect failure of the condition during the execution of the set of host instructions, means for updating state of the host computer from state of the target computer when a set of host instructions fails to execute in accordance with the speculation, and means to translate a new set of host instructions without the speculation when a set of host instructions fails to execute in accordance with the speculation.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc
Image

9. Document ID: US 6011908 A

AB: A gated store buffer including circuitry for temporarily holding apart from other memory stores all memory stores sequentially generated during a translation interval by a host processor translating a sequence of target instructions into host instructions, circuitry for transferring memory stores sequentially generated during a translation interval to memory if the translation executes without generating an exception, circuitry for indicating which memory stores to identical memory addresses are most recent in response to a memory access at the memory address, and circuitry for eliminating memory stores sequentially generated during a translation interval if the translation executes without generating an exception.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc
Image

10. Document ID: US 5958061 A

AB: Apparatus for use in a processing system having a host processor capable of executing a first instruction set to assist in running instructions of a different instruction set which is translated to the first instruction set by the host processor including circuitry for temporarily storing memory stores generated until a determination that a sequence of translated instructions will execute without exception or error on the host processor, circuitry for permanently storing memory stores temporarily stored when a determination is made that a sequence of translated instructions will execute without exception or error on the host processor, and circuitry for eliminating memory stores temporarily stored when a determination is made that a sequence of translated instructions will generate an exception or error on the host processor.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw. Desc | Image

---

☐ 11. Document ID: US 5926832 A

AB: Apparatus and a method for storing data already stored at an often utilized memory address in registers local to a host processor and maintain the data in the registers and memory consistent so that the processor may respond more rapidly when a memory address is to be accessed.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw. Desc | Image

---

☐ 12. Document ID: US 5832205 A

AB: A memory controller for a microprocessor including apparatus to both detect a failure of speculation on the nature of the memory being addressed, and apparatus to recover from such failures.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw. Desc | Image

---

☐ 13. Document ID: US 3891974 A

AB: An emulator for use in a data processing system for providing simulation of the machine wait state of the emulated central processor. A combination of hardware, firmware and software is provided to allow processing in the native mode of the data processing system while the emulated processor is in the wait state. Means are also provided for rapidly returning from the wait state to the emulation process in response to an emulator specific pending allowable interrupt. Means are further provided for indicating to the operator whether the emulated processor can in fact exit from the wait state and further means are provided for indicating to the operator the length of time the emulated processor has been in the wait state.

Terms	Documents
L15 and (floating same point)	13